IN THE CLAIMS:

Please amend the claims as follows:

1. (Amended) A program converting unit for generating a machine language instruction from a source program for [a processor] an embedded microprocessor series that manages an N-bit address while processing M-bit data, N being greater than M, N being customized depending on a program size, said program converting unit comprising:

parameter holding means for holding a data width M and a pointer width N [designated by a user], said data width M representing the number of bits of data used in the source program, [while] said pointer width N representing the number of bits of an address, said N being designated by a user depending on the program size; and

generating means for generating an instruction to manage said data width M when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width N when a variable operated by said instruction represents the address.

2. (Amended) The program converting unit of Claim 1, wherein said M is 16 and said N is [an integer] in a range of integers from 17 to 31 inclusive, said N being determined depending on the program size as follows:

N = 17, when the program size ≤ 128 Kbytes

N = 18, when the program size ≤ 256 Kbytes

N = 19, when the program size ≤ 512 Kbytes

N = 20, when the program size ≤ 1 Mbyte

N = 21, when the program size ≤ 2 Mbytes

N = 22, when the program size ≤ 4 Mbytes

N = 23, when the program size ≤ 8 Mbytes

N = 24, when the program size ≤ 16 Mbytes

N = 25, when the program size ≤ 32 Mbytes

N = 26, when the program size ≤ 64 Mbytes

N = 27, when the program size ≤ 128 Mbytes

N = 28, when the program size ≤ 256 Mbytes

N = 29, when the program size ≤ 512 Mbytes

N = 30, when the program size ≤ 1 Gbyte

N = 31, when the program size ≤ 2 Gbytes.

In Claim 3, line 3, delete "judging" (both occurrences) and insert --determining--.

13. (Amended) A program converting unit for generating a machine language instruction from a source program for [a processor] an embedded microprocessor series that manages an N-bit address while processing M-bit data, N being greater than M, N being customized depending on a program size, said program converting unit comprising:

parameter holding means for holding a data width M and a pointer width N [designated by a user], said data width M representing the number of bits of data used in the source program, [while] said pointer width N representing the number of bits of an address, said N being designated by a user depending on the program size;

generating means for generating an instruction to manage said data width M when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width N when a variable operated by said instruction represents the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, [said type being judged when said option directing means holds the user's direction for executing the overflow compensation,] said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate.

In Claim 17, line 3, delete "judging" (both occurrences) and insert --determining--.

---

20. (Amended) A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediate instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address, and one of [singed] signed and unsigned data;

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of the data, [while] said pointer width representing the number of bits of the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address; and

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate.

In Claim 24, line 3, delete "judging" (both occurrences) and insert --determining--.

27. (Amended) A program converting unit for generating a machine language instruction based on a source program for a processor that manages an N-bit address while processing M-bit data, N being greater than M, said program converting unit comprising:

syntax analyzing means for analyzing a syntax of the source program to convert the same into an intermediary language comprising intermediary instructions, and subsequently for judging whether or not each variable contained in said intermediary instructions represents data used in an address;

table generating means for generating a table for each variable in said intermediary instructions, said table holding a name together with a type of each variable, said type representing one of the data and the address, and one of [singed] signed and unsigned data;

parameter holding means for holding a data width and a pointer width designated by a user, said data width representing the number of bits of the data, [while] said pointer width representing the number of bits of the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address;

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being

generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate, wherein said generating means includes:

[judging] determining means for [judging] determining a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to [an] a memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory-access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

code generating means for generating the machine language instruction in accordance with the directions from said memory managing means, said register managing means, and said immediate managing means, and wherein

said compensate instruction generating means includes:

instruction judging means for judging an arithmetic operation instruction that will possibly cause an overflow for all the machine language instructions when said option instructing means holds the user's direction for executing the overflow compensation;

[variable] determining judging means, with respect to a variable in the arithmetic operation instructions [judged] determined by said instruction [judging] determining means, for [judging] determining an effective bit-width and whether said variable is signed or unsigned by referring to said table;

sign-extension instruction generating means for generating a compensation instruction in case of a signed variable, a logical value of a sign bit being filled into all bits higher than the effective bit-width in a register that is to store said signed variable by said sign-extension compensation instruction; and

zero-extension instruction generating means for generating a zero-extension compensation instruction in case of an unsigned variable, a logical value "0" being filled into all bits higher than the effective bit width in a register that is to store said unsigned variable by said zero-extension compensation instruction.

28. (Amended) A processor [improved in address management] being one out of an embedded processor series of processors with different address bit widths, having an address bit width which can be customized by a user in accordance with program size, comprising:

memory means for storing a program including an N-bit data arithmetic operation instruction and [both] other [N-bit and M-bit load/store] instructions operating both N-bit and M-bit data, N being greater than M;

a program counter for holding an N-bit instruction address to output the same to said memory means;

fetching means for fetching an instruction from said memory means using the instruction address from said program counter; and

executing means for executing all [N-bit] arithmetic operation instructions at N-bit length and for executing [N-bit and M-bit] other instructions [excluding] aside from the arithmetic operation instructions[,] at one of N-bit length and M-bit length;

wherein said memory means has a storage capacity equivalent of up to 2^N bytes, and whereby an N-bit address is calculated by the N-bit arithmetic operation independently of a data bit-width, said data bit-width being M.

Claim 31, line 6, delete "immediately".

Claim 35, line 6, delete "immediately".

44. (Amended) A processor for operating certain data in accordance with an instruction in a program, comprising:

a first register means for holding N-bit data;

a second register means for holding N-bit data[,];

sign-extending means for extending said M-bit data to N bits by copying an MSB of said M-bit data in a direction of an upper order, M being less than N;

zero-extending means for extending said M-bit data to N bits by copying a value "0" in a direction of an upper order;

operating means for operating an arithmetic operation in accordance with an instruction;

instruction control means for decoding an instruction to zero-extend M-bit immediate data when said M-bit immediate data are to be stored in said first register means by the decoded instruction and to sign-extend said M-bit immediate data when said M-bit immediate data are to be stored in said second register means by the decoded instruction, said zero-extended and sign-extended N-bit immediate data being outputted in one of two methods, one method being to send the extended N-bit immediate data from their respective extending means to their respective register means directly, the other being to send the same via the operating means to their respective register means, with said instruction including an indication for storing in the first register means and said instruction including an indication for storing in the second register means being of two different kinds of instruction, both having a same operation code but having different destination operands.

38

47. (Amended) A processor for operating certain data in accordance with an instruction in a program, comprising:

a first register means for holding N-bit data;

a second register means for holding N-bit data[,];

sign-extending means for extending said M-bit data to N bits by copying an MSB of said M-bit data in a direction of an upper order, M being less than N;

zero-extending means for extending said M-bit data to N bits by copying a value "0" in a direction of an upper order;

operating means for operating an arithmetic operation in accordance with an instruction;

instruction decoding means for decoding an instruction in the program to detect a first type instruction and a second type instruction, said first type instruction including an instruction to store M-bit immediate data into said first register means, said second type instruction including an instruction to store said M-bit immediate data into said second register means; and

control means for outputting said M-bit immediate data to said zero-extending means when the first type instruction is detected, and for outputting said M-bit immediate data to said sign-extending means when the second type instruction is detected, said zero-extended N-bit immediate data and sign-extended N-bit immediate data being outputted in one of two methods, one method being to send the extended N-bit immediate data from their respective extending means to their respective register means directly, the other being to send the same via the operating means to their respective register means, with said first-type instruction and said second-type instruction ~~both~~ having a same operation code but having different destination operands.

42

51. (Amended) A data processing method for executing an instruction that includes an [instruction] operation code to store M-bit immediate data in an N-bit first register and an N-bit second register, both M and N being integers, with [while] M being less than N, said method comprising the steps of:

decoding [an] the instruction for selecting one of the first register and second register in accordance with [a] an operand of the decoded instruction;

zero-extending said M-bit immediate data to N bits when said decoded instruction designates the first register, and sign-extending said M-bit immediate data to N bits when said decoded instruction designates the second register; and

storing extended N-bit immediate data to the designated register.

45

54. (Amended) A processor [for executing a program including an N-bit data arithmetic operation instruction, M-bit and N-bit load/store instruction, M being less than N, a conditional branch instruction, a data-transfer instruction with an external memory, and an instruction having immediate data, said processor comprising:] being one out of an embedded processor series of processors with different address bit widths, having an address bit width which can be ~~customized~~ input by a user in accordance with program size, comprising:

memory means for storing a program including an N-bit data arithmetic operation instruction and other instructions operating both N-bit and M-bit data, N being greater than M, as well as for storing a program including conditional branch instructions, transfer instructions for external memory and instructions using immediate data;

a first register means including a plurality of registers for holding N-bit data;

a second register means including a plurality of registers for holding N-bit data;

a program counter for holding an N-bit instruction address to output the same to said memory means;

fetching means for fetching an instruction from an external memory using the instruction address from said program counter;

instruction decoding means for decoding a fetched instruction;

executing means for executing all [N-bit] arithmetic operation instructions at N-bit length and for executing [N-bit and M-bit] instructions operating both N-bit and M-bit data excluding the arithmetic operation instructions[,];

a plurality of flag storing means, each for storing a corresponding flag group changed in response to different bit-widths data in accordance with an execution result of said executing means;

flag selecting means for selecting a certain flag group from said plurality of flag storing means in accordance with a conditional branch instruction decoded by said instruction decoding means;

branch judging means for judging whether a branching is taken or not with a reference to a flag group selected by said flag selecting means;

sign-extending means for extending M-bit data to N bits by copying an MSB of said M-bit data in a higher order;

zero-extending means for extending M-bit data to N bits by filling a value "0" in a higher order;

compensation instruction control means for compensating contents of said first register means and said second register means using said sign-extending means and said zero-extending means in accordance with a compensation instruction inserted [immediately] after a machine language instruction for an arithmetic operation that will possibly cause an overflow, said machine language instruction being decoded by said instruction decoding means;

external-access-width control means for outputting bit-width information for transmission data in accordance with a type of said register means to which a register indicated by register information belongs, said register information indicating one of said first and second register means;

external-access executing means for executing a data transfer between the register and an external memory in accordance with said register information and bit-width information; and

immediate control means for outputting M-bit immediate data to said zero-extending means when a decoded instruction includes an instruction to store said M-bit immediate data in said first register means, and for outputting said M-bit immediate data to said sign-extending means when a decoded instruction includes an instruction to store said M-bit in said second register means, said zero-extended and sign-extended immediate data being sent to said first and second register means respectively in two methods, one being to send the same directly to their respective register means and the other being to send the same via said executing means,

wherein said memory means stores a program of a size which is up to 2^N bytes.

Please add the following newly-drafted Claims 56 and 57:

56. A program converting unit for generating a machine language instruction from a source program, the machine language program being generated for a selected microprocessor in an embedded microprocessor series comprising a plurality of microprocessors, each of the plurality of microprocessors being able to process M-bit data and having a different address bit width N, said program converting unit comprising:

parameter holding means for holding a data width M and a selected pointer width N, N and M being integers greater than zero and N being greater than M,

said data width M representing a bit-width of data used in the source program to be converted,

said pointer width N representing an address bit-width to be used with the converted machine language program and being set by a user, depending on an estimated size of the object program after conversion, in order to identify the selected microprocessor in the embedded microprocessor series; and

generating means for generating an instruction to manage said data width M when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width N when a variable operated by said instruction represents the address.

48

57. A program converting unit for generating a machine language instruction from a source program, the machine language program being generated for a selected microprocessor in an embedded microprocessor series comprising a plurality of microprocessors, each of the plurality of microprocessors being able to process M-bit data and having a different address bit width N, said program converting unit comprising:

parameter holding means for holding a data width M and a selected pointer width N, N and M being integers greater than zero and N being greater than M,

said data width M representing a bit-width of data used in the source program to be converted,

said pointer width N representing an address bit-width to be used with the converted machine language program and being set by a user, depending on an estimated size of the object program after conversion, in order to identify the selected microprocessor in the embedded microprocessor series;

generating means for generating an instruction to manage said data width M when a variable operated by said instruction represents the data, and for generating an instruction to manage said pointer width N when a variable operated by said instruction represents the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said compensation instruction being generated when an effective bit width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate.